

AD-A093 744

RAND CORP SANTA MONICA CA

F/6 22/4

MEASURING THE IMPACT OF LAUNCH OPERATIONS UPON SATELLITE EFFECT--ETC(U)

JUN 80 B E KRELL

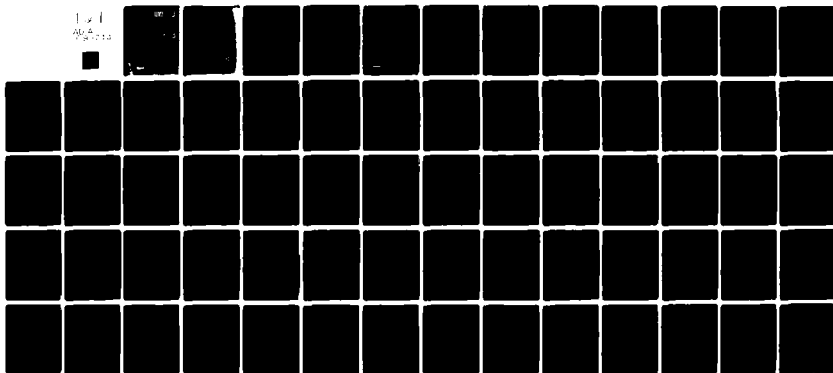
F49620-77-C-0023

UNCLASSIFIED

RAND/N-1528-AF

NL

1 of 1
AD-A093 744



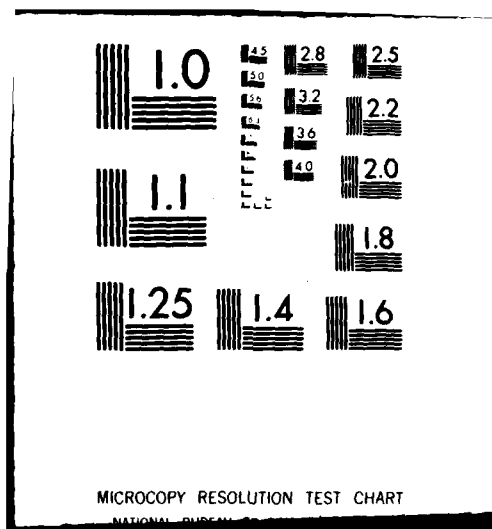
END

DATE

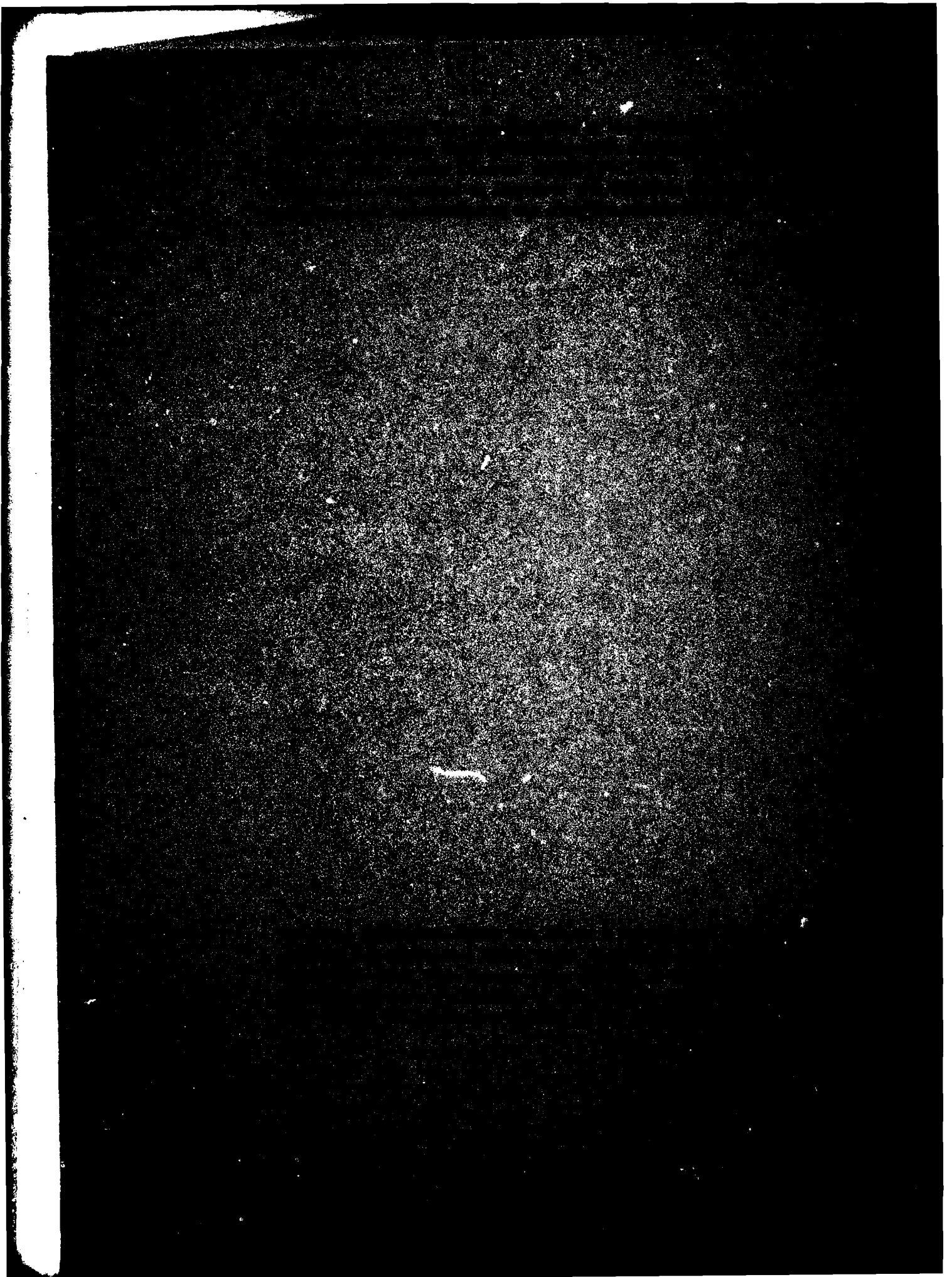
FILED

2-81

DTIC



AD A093744



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RAND/N-1528-AF	2. GOVT ACCESSION NO. AD-A093744	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Measuring the Impact of Launch Operations upon Satellite Effectiveness.	5. TYPE OF REPORT & PERIOD COVERED (9) Interim rept. /	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bruce E. Krell	8. CONTRACT OR GRANT NUMBER(s) (15) F49620-77-C-0023	9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11)
10. PERFORMING ORGANIZATION NAME AND ADDRESS The Rand Corporation 1700 Main Street Santa Monica, CA 90604	11. CONTROLLING OFFICE NAME AND ADDRESS Requirements, Programs & Studies Group (AF/RDQM) Ofc, DCS/R&D and Acquisition Hq USAF, Washington, DC 20330	12. REPORT DATE June 1987
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 68	14. SECURITY CLASS. (of this report) UNCLASSIFIED	15. NUMBER OF PAGES 61
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Distribution: Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) No Restrictions		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Satellite Networks Launch Vehicles Systems Analysis Space Shuttles Procurement Simulation Mathematical Models		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) See Reverse Side		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 296600
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

✓ Procurement decisions regarding Department of Defense satellite systems are usually based upon the assumption of a reliable and dependable launch system. Attempts to relate disruptions of launch operations to numbers of satellites on-orbit are often in the form of discrete-event simulations (requiring significant numbers of data inputs and large amounts of computer time). (The approach which is adopted in this study is a form of continuous simulation (as opposed to discrete-event simulation), an approach which has traditionally been limited to analysis of systems in the physical sciences. The continuous simulation methodology requires significantly fewer inputs than discrete-event simulations and is amenable to being implemented on a programmable, hand-held calculator, saving expensive computer and data-gathering time. Several examples are provided which demonstrate how continuous simulation may be used to model disruptions in launch capability and various satellite operational philosophies. 61 pp. Ref. (Author)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

A RAND NOTE

MEASURING THE IMPACT OF LAUNCH OPERATIONS UPON SATELLITE EFFECTIVENESS

Captain Bruce E. Krell

June 1980

N-1528-AF

Prepared For

The United States Air Force

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Avail and/or	
Dist	Special
A	



PREFACE

Procurement decisions regarding Department of Defense satellite systems are usually based upon the assumption of a reliable and dependable launch system. Attempts to relate disruptions of launch operations to numbers of satellites on-orbit often take the form of discrete-event simulations (requiring significant numbers of data inputs and large amounts of computer time). The purpose of this Note is to present an alternative method for evaluating how launch operations affect orbital status. The method is an extension of a previous modeling effort explained in Rand Note N-1295-AF, Cost-Effectiveness Measures of Replenishment Strategies for Systems of Orbital Spacecraft.

A distinct difference exists, however, between the present Note and the previous one. Because variations in launch activities are likely to be transient phenomena, the modeling approach adopted here must be capable of reflecting the orbital status of a satellite system over time. The transient-state-analysis approach that is adopted is a form of continuous simulation (as opposed to discrete-event simulation), an approach traditionally limited to the analysis of systems in the physical sciences.

Extension of the continuous-simulation approach to operational phenomena should provide a whole new perspective to analysts dealing with functional systems and organizations. This methodology requires significantly fewer inputs than discrete-event simulations and can be implemented on a programmable, hand-held calculator, saving expensive computer and data-gathering time.

This work was done under the Project AIR FORCE study effort "Space Shuttle Issues."

At the time this Note was prepared, the author was assigned by the Air Force Space Division to The Rand Corporation as a Research Associate.

P081 859

SUMMARY

The successful operation of a space launch system, such as the space shuttle, is vital to the maintenance and improvement of satellite operations. This Rand Note demonstrates that a reasonable method exists for quantifying this dependence relationship, laying the groundwork for a cost-effectiveness analysis of alternative backup launch systems.

The methodology employed is an extension of the approach described in the author's Note N-1295-AF, Cost-Effectiveness Measures of Replenishment Strategies for Systems of Orbital Spacecraft, December 1979. In that document, changes in the status of a satellite orbital configuration were represented by a system of ordinary, linear, differential equations. These equations were then used to derive steady-state cost-effectiveness measures. The present Note formulates the generalized version of these differential equations, gives examples of appropriate time-dependent equation coefficients, and shows how continuous simulation (numerical integration) is employed to obtain useful comparisons from the equations. The analysis can now reflect a variety of transient phenomena in the context of spacecraft and launch system interaction.

The methodology employed here has several benefits. As explained within the text, very few pieces of data are required to reflect the impact of transient phenomena, yielding high informational content. Moreover, concepts of continuous simulation (normally applied to physical mechanics problems) have been extended to the study of behavioral/sociological (queueing) systems.

Specific examples include a comparison of the impact of power cycling and the risks associated with failure to provide backup launch systems to the space shuttle. In these examples, the appropriate differential equation coefficients are explicated (as a function of satellite and launch system operational characteristics), and the appropriate data are employed in a continuous simulator. The actual simulation code is documented in the Appendixes.

CONTENTS

PREFACE	111
SUMMARY	v
Section	
1. INTRODUCTION	1
1.1 The Scenario for Launch System Operation	2
1.2 Some Measures of Satellite System Performance	3
1.3 A General Model of Orbital/Launch Operations	5
2. COMPUTATIONS FOR MEASURING THE IMPACT OF LAUNCH SYSTEMS ...	6
2.1 Closed Form Solutions	7
2.2 The Diffusion Approximation Approach	10
2.3 Continuous Simulation of Satellite/Launch Operations	12
2.4 The SPACE Model	13
2.4.1 Required Inputs	14
2.4.2 The Algorithms Used	15
2.4.3 Modeling of Orbital/Launch Activities	16
3. THE INFINITE INVENTORY CASE	19
3.1 Closed Form Solution	21
3.2 Comparison with SPACE Model	23
3.3 Limitations of the Simple Example	25
4. SPECIAL CASES	26
4.1 Satellite Power Cycling	26
4.2 Disruptions in Launch Operations	30
4.3 The SPACE Model and Cost-Effectiveness	34
5. CONCLUSIONS	37
Appendix	
A. USER INSTRUCTIONS	39
B. MODEL DOCUMENTATION	40
C. "SPACE" MODEL SOFTWARE	42
REFERENCES	61

1. INTRODUCTION

Satellites are placed in various orbits in space for a whole range of functions, including communications, weather monitoring, etc. When satellites on-orbit fail, they must be replenished. Even if this replenishment is accomplished via an orbital spare, the requirement for a reliable and dependable launch system cannot be avoided (even the spare has to be launched). Thus, the existence and successful operation of a space launch system is vital to the maintenance of an orbital configuration of satellites. The goal of this Note is to demonstrate that a reasonable method exists for quantifying this dependence between satellite operations and launch operations.

In order to realistically evaluate the launch/orbital relationships, the approach employed must be able to reflect the impact of variations in operational activities. For instance, a satellite system is often procured as a single-lot buy. After the initial procurement is launched, no more launches occur. Thus, the relationship between launches and orbital operations varies depending upon the phase of the procurement process. During procurement, a launch schedule dominates the relationship; after procurement, the fact that no launches occur must be reflected. These transient phenomena are usually implemented in the form of a discrete-event simulation [1].

However, a discrete-event simulator usually requires a large data set, significant execution time, and disproportionate development time. The approach employed here is significantly different. The relationship between launches and orbital activities is represented as a system of simultaneous differential equations. For a given set of initial conditions, the equations can be numerically integrated over time to reflect orbital status in response to transient launch phenomena. In recent years, this numerical integration approach has been applied to physical systems and is referred to as continuous systems simulation. As will be demonstrated, a variety of changes in launch conditions and orbital considerations are handily modelled through the appropriate choices of the coefficients in the differential equations.

The appendixes to this Note contain the required user documentation for a continuous simulator of satellite systems. Some of the applications of this simulator appear in Sections 3 and 4. Moreover, an example is presented which compares a closed form analysis to the simulated results for a severely restricted case. An additional, useful section demonstrates ways in which the simulator outputs may be employed in making policy decisions regarding launch system procurement.

1.1 The Scenario for Launch System Operation

Space launch systems exist for the primary purpose of placing satellites into an orbital plane. A space launch system is defined as the set of operational systems required to place a satellite in orbit, which includes the launch mechanism itself and all of its ground support equipment. An assumption is made that launches occur on a regularly scheduled basis so that summary performance characteristics of the launch system may be defined.

Launch systems may be classified as reusable or expendable. For reusable systems, one may define the turn-around-time (TAT) as the average time from landing to relaunch. The performance of expendable systems may be succinctly described by the minimum time to launch (MTTL). The MTTL is defined as the average time required to prepare and to launch assuming a launch vehicle is available in inventory. In either case, a series of launches (regardless of the vehicle source) may be summarized by the mean-time-between-launches (MTBL). In this case, the meaning is obvious.

The goal of operating a launch system focuses upon providing a service to any satellite program. In most cases, the satellite program desires to maintain a certain number of satellites on-orbit in a certain orbit (in order to perform a specified mission). Thus, the performance of the satellite system can be measured by the average number of satellites on-orbit. An additional performance measure for satellite systems is the availability of the system. A satellite system is said to be available if a certain specified number of satellites is on-orbit. (This minimum number required varies across satellite programs depending on the mission of the program.)

Discussions below indicate that both the average number of satellites and the system availability are derived from the same set of probabilistic data. Thus, measuring one of the performance characteristics necessarily implies knowledge of the others.

This Note relates the performance of satellite systems (as measured by the average number of satellites on-orbit) to the performance of launch systems (as measured by one of TAT, MTTL, or MTBL). The model developed is general enough to permit comparisons of the impact of a variety of transient phenomena (launch disruptions, etc.). The model is relatively easy to implement and to modify and requires minimal amounts of data (relative to large-scale, discrete-event simulators).

1.2 Some Measures of Satellite System Performance

In order to measure satellite effectiveness on a transient basis, a few simple extensions of a previous work are necessary. In Rand N-1295-AF [1], a state-space, Markov process approach was employed to assess the on-orbit availability of satellite systems. The logical analogues of the Markov process performance measures are presented in this section.

For purposes of this discussion, the following notation is defined:

n = number of operational satellites on-orbit;

M = maximum number of satellites on-orbit;

$\lambda_n(t)$ = expected failure rate per unit time of an individual satellite, as a function of the number of satellites on-orbit (n) and the specific time of the operational cycle (t);

$\mu_n(t)$ = expected replenishment rate per unit time of an individual satellite, as a function of the number of satellites on-orbit (n) and the specific time of the operational cycle (t);

$\{\lambda_n(t)\}$ = the set of all failure rates;

$\{\mu_n(t)\}$ = the set of all replenishment rates;

$A(t)$ = availability of a satellite system at time t ;

$E[n|t]$ = average number of satellites on-orbit at time t ;

$P_n(t)$ = probability that n satellites are on-orbit at time t .

The notations for failure and replenishment rates represent the most general form and are meant to reflect the transient nature of various operational concerns. In fact, a single member of $\{\lambda_n(t)\}$ or $\{\mu_n(t)\}$ may itself be a set of functions. Specific examples are presented in Sections 3 and 4.

The performance of the satellite system may now be measured as follows:

$$A(t) = P_N(t) + \dots + P_M(t)^*$$

$$E[n|t] = \sum_{n=0}^M nP_n(t)$$

where

$$P_n(t) = f[\{\lambda_n(t)\}; \{\mu_n(t)\}]$$

and

$$\{\mu_n(t)\} = g[\text{TAT or MTTL or MTBL}]$$

$$\{\lambda_n(t)\} = h[\text{MMD}]^{**}$$

Subsequent sections give examples of how f , g , h may be numerically or explicitly determined. Thus, the performance of the satellite system can be directly related to the performance of the launch system. Moreover, since the form of $\{\mu_n(t)\}$ is general, a variety of launch operational phenomena can be analyzed.

* Assumes a minimum of $N \leq M$ satellites are required for availability.

** MMD = Mean Mission Duration = Average measure of satellite lifetime under the satellite reliability function.

As described earlier, the above discussion demonstrates that both availability and expected number on-orbit can be computed from knowledge of the $P_n(t)$. Thus, once the latter have been computed, one can present the information in either form. For the remainder of this Note, the result of any analysis is presented in the form of $E[n|t]$.

1.3 A General Model of Orbital/Launch Operations

Using the terminology described above, a set of conditions can be described which characterizes the $P_n(t)$. This set of conditions takes the form of a system of ordinary differential equations with corresponding boundary conditions. These equations with boundary conditions form the general model of orbital/launch operations.

Using a state-space approach, as in [1], one can demonstrate that the $P_n(t)$, upon which satellite performance is measured, must satisfy the following conditions:

$$(1.1) \quad \frac{dP_0(t)}{dt} = -\mu_0(t)P_0(t) + \lambda_1(t)P_1(t) \quad n = 0$$

$$(1.2) \quad \frac{dP_n(t)}{dt} = -[n\lambda_n(t) + \mu_n(t)]P_n(t) + \mu_{n-1}(t)P_{n-1}(t) \\ + (n+1)\lambda_{n+1}(t)P_{n+1}(t) \quad 0 < n < M$$

$$(1.3) \quad \frac{dP_M(t)}{dt} = -M\lambda_M(t)P_M(t) + \mu_{M-1}(t)P_{M-1}(t) \quad n = M$$

$$(1.4) \quad P_n(0) = \begin{cases} 1 & \text{if } n = a \\ 0 & \text{if } n \neq a \end{cases}$$

where a = number on-orbit when the system state is initialized.

This system of ordinary differential equations demonstrates how the $P_n(t)$ change over time, as a function of launch system operations--

reflected by the appearance of $\{\mu_n(t)\}$ throughout the equations. A close examination of the equations reveals that launches are conditioned upon failures during previous time periods.

For example, in (1.2) the pairing of $\mu_{n-1}(t)$ with $P_{n-1}(t)$ says that a failure must have occurred previously for a launch to occur. Moreover, the pairing of $\lambda_{n+1}(t)$ with $P_{n+1}(t)$ is an attempt to capture the residual lifetime remaining at any time t . Thus, the general form of the model effectively replicates the day-to-day activities of an orbital system and its launch support system (on an infinitesimal basis). The details for deriving (1.1) to (1.4) are a simple extension of the approach employed in Section 2 of [1].

While the equations above do characterize the transient behavior of the $P_n(t)$, they need to be solved. The next section discusses a variety of methods and conditions for finding solutions to this general system of equations.

2. COMPUTATIONS FOR MEASURING THE IMPACT OF LAUNCH SYSTEMS

Solving the equations for the general model, as presented in Section 1.3, is not a simple task. The inclusion of the parameter M and the parameter sets $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ severely inhibits the obtaining of solutions to the equations. Several alternative approaches are discussed in this section. In certain restrictive situations, solutions are available. Techniques for obtaining these solutions, examples, and discussions of the shortcomings of each situation are presented.

The first example deals with finding a closed form solution. In this case, the Laplace Transform approach is used to solve the system of equations. A second approach is to employ a diffusion approximation. The system of equations is replaced by a single partial-differential equation with border conditions. Finally, continuous simulation employing simple numerical integration is accomplished. This latter approach is the most general and a resultant simulation model is presented. The remainder of this Note in Sections 3 and 4 deals with the effective usage of the simulation model described.

2.1 Closed Form Solutions*

As the first example, the system of equations is solved by Laplace Transform. In order to obtain a closed form solution by this method, several simplifying assumptions must be made.

Consider a satellite system consisting of a single satellite. If an infinite inventory of satellites exists and an infinite number of launch vehicles is available, then this case may be summarized by

$$(2.1) \quad M = 1$$

$$(2.2) \quad \lambda_n(t) = \lambda = \frac{1}{\text{MMD}} \quad \forall n, t$$

$$(2.3) \quad \mu_n(t) = \mu = \frac{C}{\text{MTBL}} \quad \forall n, t$$

where C = number satellites launched per launch vehicle,

MMD = mean mission duration = average lifetime of a satellite,

MTBL = mean time between launches.

Then, conditions (1.1) to (1.4) reduce to

$$(2.4) \quad \frac{dP_0(t)}{dt} = -\mu P_0(t) + \lambda P_1(t)$$

$$(2.5) \quad \frac{dP_1(t)}{dt} = -(1)\lambda P_1(t) + \mu P_0(t)$$

$$(2.6) \quad P_n(t) = \begin{cases} 1 & n = 0 \\ 0 & n = 1 \end{cases}$$

Note that for this case, the differential equations are linear in form. The solution process is significantly eased since only two linear equations are required.

* At a first reading, Sections 2.1 and 2.2 may be skipped without loss of continuity.

The Laplace Transform of $P_n(t)$ is defined to be the function $\phi_n(\theta)$ such that

$$\phi_n(\theta) = \int_0^{\infty} e^{-\theta t} P_n(t) dt$$

Using integration by parts on the above expression yields

$$\int_0^{\infty} e^{-\theta t} \frac{dP_n(t)}{dt} dt = \phi_n(\theta) - P_n(0)$$

Applying the transform operators defined above converts the simple linear differential equations in (2.4) to (2.6) to the following:

$$(2.7) \quad (\theta + \mu)\phi_0(\theta) - \lambda\phi_1(\theta) = 1$$

$$(2.8) \quad -\mu\phi_0(\theta) + (\theta + \lambda)\phi_1(\theta) = 0$$

Solving these two equations simultaneously yields the Laplace Transforms in closed form:

$$(2.9) \quad \phi_0(\theta) = \frac{\theta + \lambda}{\theta(\theta + \lambda + \mu)}$$

$$(2.10) \quad \phi_1(\theta) = \frac{\mu}{\theta(\theta + \lambda + \mu)}$$

Note that the denominator of each is a polynomial of order 2, i.e., of order $M + 1$. In order to use the method of the residues to invert these expressions, one must find the real roots of this $(M + 1)$ st order polynomial. For this example, one can easily determine that there is one root (or pole) at $\theta = 0$ with order $m = 1$ and one pole at $\theta = -\lambda - \mu$ with order $m = 1$.

According to the method of the residues, if a is a pole of $\phi_n(\theta)$ with order m , then the residue of $\phi_n(\theta)$ at the pole $\theta = a$ is defined as

$$r_a = \frac{1}{(m-1)!} \lim_{\theta \rightarrow a} \left\{ \frac{d^{m-1}}{d\theta^{m-1}} [(\theta - a)^m \phi_n(\theta) e^{\theta t}] \right\}$$

and $P_n(t)$ can be found by

$$P_n(t) = \sum_{a="a \text{ pole}"} r_a$$

If the poles and orders above are inserted into the residue equations, the resultant probabilities are

$$(2.11) \quad P_0(t) = \frac{\lambda}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} e^{-(\mu + \lambda)t}$$

$$(2.12) \quad P_1(t) = \frac{\mu}{\mu + \lambda} - \frac{\mu}{\mu + \lambda} e^{-(\mu + \lambda)t}$$

Assuming that the satellite is available if the single satellite is on-orbit, and using the definitions for availability and expected number on-orbit, one finds that

$$(2.13) \quad A(t) = P_1(t) = \frac{\mu}{\mu + \lambda} - \frac{\mu}{\mu + \lambda} e^{-(\mu + \lambda)t}$$

$$(2.14) \quad E[n|t] = 0P_0(t) + 1P_1(t) = P_1(t) = A(t)$$

Recall that $\mu = 1/\text{MTBL}$ so that, in this simple case, both availability and number on-orbit are related to the performance of the launch system (represented by MTBL).

Unfortunately, this simple example is not of great use because of the inherent assumptions and simplicity. First, an infinite number of satellites is not usually procured. The assumption regarding

the launching of these satellites is that launches occur on a regularly scheduled basis. Thus, this assumption eliminates any consideration of transient phenomena in the launch process (one of the primary goals of this Note). Moreover, solutions (2.13) and (2.14) are only appropriate for a satellite system consisting of a single satellite. Many satellite systems consist of procurements of size $M > 1$. Recall that inverting the Laplace Transforms required finding the poles (or roots of the denominator) of an $(M + 1)$ st order polynomial. For $(M + 1)$ any larger than 2, the algebraic expressions for finding the roots of a polynomial are either quite algebraically difficult or are nonexistent $[(M + 1) \geq 5]$. Additionally, the results of applying Laplace Transforms to $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ are difficult to predict since the latter represent general forms. Unless the actual forms are known, one cannot ascertain the impact of applying the transform operators. For these reasons, this example provides severely limited information for the general approach.

2.2 The Diffusion Approximation Approach

Another somewhat successful approach is to approximate the solution to (1.1) to (1.4) by replacing the system of equations with a single partial differential equation. In order to use this approach, the discrete variable n (number of satellites on-orbit) must be replaced by a continuous variable denoted by x . This technique is referred to by Gross and Harris [2] as a diffusion approximation. The solution to the single equation with boundary conditions is extremely difficult to obtain. Thus, the solution is not presented below. However, the appropriate use of the solution is indicated.

Assume that one can replace n by x in equation (1.2) and in the parameter sets for failure and replenishment rates. Then, (1.2) can be written in the following form (including boundary conditions):

$$(2.15) \quad \frac{\partial P(x, t)}{\partial t} = -[x\lambda(x, t) + \mu(x, t)]P(x, t) + \mu(x - 1, t)P(x - 1, t) \\ + (x + 1)\lambda(x + 1, t)P(x + 1, t)$$

$$(2.16) \quad P(x, t) \geq 0$$

$$(2.17) \quad \int_0^M P(x, t) dx = 1$$

$$(2.18) \quad \lim_{t \rightarrow 0} P(x, t) = 0$$

$$(2.19) \quad \lim_{x \rightarrow 0} P(x, t) = \lim_{x \rightarrow 0} \frac{\partial P(x, t)}{\partial t} = 0 \quad (\forall t)$$

$$(2.20) \quad \lim_{x \rightarrow M} P(x, t) = \lim_{x \rightarrow M} \frac{\partial P(x, t)}{\partial t} = 0 \quad (\forall t)$$

The last two equations ensure that the resultant solution $P(x, t)$ contains all probability within the range $0 \leq X \leq M$, (X =no. satellites).

If the solution to equation (2.15) with boundary conditions (2.16) to (2.20) can be found, the $P(x, t)$ can be employed to approximate the availability and the expected number of satellites on-orbit over time. Since $P(x, t)$ approximates the number on-orbit as a continuous variable x , then the computations must be performed as follows:

$$(2.21) \quad A(t) = \int_N^M P(x, t) dx$$

$$(2.22) \quad E[x|t] = \int_0^M xP(x, t) dx$$

The actual solution $P(x, t)$ is not presented here because the forms of $\{\lambda_n(x)\} \rightarrow \{\lambda(x, t)\}$ and $\{\mu_n(t)\} \rightarrow \{\mu(x, t)\}$ must be known in order to solve for $P(x, t)$. Gross and Harris [1] give an example for which $\mu_n(t) = \mu = \frac{1}{MTBL} \forall n, t$ and $\lambda_n(t) = \lambda = \frac{1}{MND} \forall n, t$ and $M \rightarrow \infty$. If an infinite number of satellites are procured by a program and all of

these are launched on a continuing, repetitive basis, then the solution for $P(x, t)$ given by Gross and Harris [2] is applicable. Unfortunately, these assumptions are unrealistic and contradictory to the goal of reflecting the impact of transient launch phenomena on satellite performance. Therefore, this approach fails to have utility.

2.3 Continuous Simulation of Satellite/Launch Operations

When closed form or analytic solutions do not exist, one usually resorts to simulation. Since solutions to the original conditions on $P_n(t)$ [(1.1) to (1.4)] do not readily admit to solution, the simulation approach seems the best possible approach here also. In most cases, discrete-event simulation is used. However, the simulation approach advocated here is of a different type, referred to as continuous simulation. A brief comparison of the two simulation techniques is described below in an attempt to show why continuous simulation is preferred.

The typical discrete-event simulator generates sample events of various activities drawn from the underlying stochastic processes governing the behavior of the activities. The historical sequences of activities are maintained, creating a large statistical data base. Using this data base, one can generate a large variety of outputs. Moreover, since individual events are being generated, the discrete simulator can model interactions of events to significant detail. Unfortunately, this attention to detail requires significant computer execution time, extensive memory space, and a largely unjustified amount of preparation and model construction time. If changes are required in the model, extensive reprogramming, debugging, and testing are usually necessary. The discrete approach is detailed but inflexible.

If continuous simulation is employed, the interactions of the underlying processes are succinctly modelled as a single or a system of differential equations with corresponding boundary conditions. The technique of continuous simulation simply consists of numerically integrating the differential equations. The simplicity of this approach has several distinct advantages. Very few inputs are required; significantly less memory space and execution time are necessary.

Variations in interactions of processes may be easily reflected by changing the coefficients of the differential equations--which may be limited to a single subroutine if the simulator is constructed correctly. This approach is so simple that the whole simulator can be fairly easily implemented on a medium-size programmable calculator with printing capability.

The key detriment to continuous simulation is the limitation on obtainable outputs. Only variables dependent on $P_n(t)$ (for this model) can be computed. As shown above, both availability and average number on-orbit over time require knowledge of $P_n(t)$. Thus, the continuous simulation approach provides maximum utility for the smallest data input requirements and is the focus of the remainder of this Note. For examples of discrete-event simulators used to analyze satellite systems, see Reference [1].

2.4 The SPACE Model

Implementing the numerical integration process required by continuous simulation is itself a relatively simple task. However, the specific usage of the computer model requires the ability to alter the underlying models quite rapidly. For this reason, a modular construction is employed. Each routine exists to perform a specific task or activity. One main routine and five subroutines form a continuous simulation model called SPACE.

The MAIN routine is a few simple lines which call the other routines and establish the major logic of the model. The functions of the other routines may be briefly described as follows:

1. READ--reads the input data.
2. INTEG--performs the numerical integration process.
3. WRITE--writes the outputs $[E[n|t], [P_n(t)]$, to the appropriate disk files.
4. AVG--computes the average number at time t .
5. FNS--contains the implementation of equations (1.1) to (1.4) which form the rates of change of the probabilities.

The exact software appears in Appendix C. As one can see, the portion of the programming to be modified is strictly confined to the subroutine entitled FNS. Moreover, the latter routine is invoked at every time instant during the integration process. This allows the implementation of time-dependent coefficients $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$. Examples appear throughout Section 4.

2.4.1 Required Inputs

The simplicity of the numerical integration approach dictates only 10 inputs. Five of these deal with the basic parameters describing orbital and launch operations. Most of these will appear to be obvious in view of the discussion in previous sections. The remaining inputs specify the mechanics of the simulation process. In order to use the SPACE Model, one needs to perform the steps described in Appendix A. Since the model operates interactively, execution must occur under IBM's Time Sharing Option (TSO). In other words, the user is prompted for the inputs during execution at a terminal.

Describing the basic parameters of launch/orbital operations requires one to input the following variables:

1. $P\phi$ --probability that a booster fails.
2. MMD--mean mission duration = average life of a satellite
(= 1/average number failures per unit time = 1/failure rate).
3. C--number satellites/launch vehicle.
4. MU--average number launches/unit time = (1/Mean Time Between Launches):(1/MTBL) = launch rate.
5. M--maximum number of satellites on-orbit (i.e., number satellites procured).

In Section 4 examples are given which demonstrate that almost all members of the sets $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ can be expressed as functions of the above basic parameters.

Parameters for continuous simulation are significantly fewer in number than required by discrete-event simulators. The simulation parameters required are as follows:

6. LT--length of the simulation time interval (in time units consistent with MTBL and MMD above).
7. NS--number of subintervals within time interval LT, (NS > 100).*
8. T \emptyset --the time for initialization of the simulation. If T \emptyset > \emptyset , a special integration step computes P $_n$ (T \emptyset).
9. PMIN--a stopping criterion which halts the simulation process if P $_0$ (t) is smaller.
10. TMAX--another stopping criterion which is interpreted as the maximum number of increments of size LT for simulating.

In some instances, these parameters may also be employed in determination of members of $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$. See Section 4 for specific examples.

2.4.2 The Algorithms Used

Equations (1.1) to (1.4) form a system of ordinary differential equations which are quite simple to integrate. For this reason, the scheme adopted for numerical integration is the most simple available--Euler's Method.

In terms of the variables used in this Note, Euler's Method may be described in the following manner:

$$(2.23) \quad P(n)_{t+D} = P(n)_t + \left\{ \left[\frac{dP(n)}{dt} \right]_t \cdot D \right\} \quad n = 1, \dots, M+1$$

where

$$(2.24) \quad P(n)_t = P_{n-1}(t) \quad n = 1, \dots, M+1$$

$$(2.25) \quad D = (LT/NS) = \text{length of one subinterval within LT}$$

$$(2.26) \quad \frac{dP(n)}{dt}_t = \text{determined from (1.1) to (1.4)}$$

*Testing by the author revealed that using NS = 100 provided the maximum trade between accuracy and computational time.

Equation (2.23) is the basic form of Euler's Method expressed in terms of the variables appropriate to this Note and is easily verified by any standard text on differential equations. Since the simulator is written in FORTRAN, $P_0(t)$ cannot be recorded (zero subscripts are not allowed). Thus, Equation (2.24) is necessary to translate the program vector $P(n)_t$ into the state variable measure $P_n(t)$. The remaining equations are obvious.

The notation $P(n)_t$ is adopted because all probabilities values over time are not permanently stored in main memory. (They are, however, stored in a disk file. See Appendix A regarding execution instructions and descriptions of disk files.) Only the current and previous values-- $P(n)_{t+D}$ and $P(n)_t$ --are stored.

From (2.23), the performance measures cited earlier are easily computed:

$$(2.27) \quad A(t) = \sum_{n=N+1}^{M+1} P(n)_t$$

$$(2.28) \quad E[n|t] = \sum_{n=1}^{M+1} (n-1)P(n)_t$$

Again, the FORTRAN vector subscripting problem results in slight variations in the formulas. In fact, (2.27) and (2.28) represent approximations of the values of $A(t)$ and $E[n|t]$. A comparison of this approximation with actual values (for a limited, special case) appears in Section 3.

2.4.3 Modelling of Orbital/Launch Activities

Choosing the coefficients in (1.1) to (1.4) for the sets $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ corresponds to modelling the day-to-day activities for the interacting launch and orbital systems (on an infinite basis). These time dependent coefficients are then computed at each time step of the integration process described by (2.23) to (2.26). Updating of these coefficients and the rates of change occurs in subroutine FNS.

This subroutine consists of six distinct blocks. A heading block identifies the beginning of the subroutine and consists of a single line--SUBROUTINE FNS. The type declaration block specifies both the type and dimension of each variable employed. Any version of subroutine FNS must contain this block in the following form:

```
INTEGER M,NS
REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX
REAL P(75),E(500),P1(75),T(500),T2,F,MUT
INTEGER I,J,K
```

The block or segment following the type declaration block is called the COMMON block. This segment determines the variables passed between the program SPACE subroutines. This block is also mandatory and must exactly follow the type declaration segment. Construction of the COMMON block is as follows:

```
COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX
COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I
```

Descriptions of the variables in the type declaration and COMMON blocks are found in Appendix B.

The latter blocks deal with the mechanics of making software modules compatible. The final two blocks establish the modelling process. Block three computes the members of the parameter/coefficient sets $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ while the fourth block updates the rates of change according to (1.1) to (1.4). In the current application, the coefficient set block is used to determine two variables representing the current member of $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$. The variables are described as follows:

$$(2.29) \quad L = f(MMD) \quad [-\lambda_n(t)]$$

$$(2.30) \quad MUT = g(C, MU) \quad [-\mu_n(t)]$$

where $MU = C/MTBL$.

Several other parameters may be of interest in determining the form of f, g to be used. These parameters are defined below:

1. $T1$ --the beginning of the current time interval of simulation.
2. D --the size of a time subinterval for which the rate of change update is being computed.
3. J --a counter for the number of subintervals (with a single time interval).
4. I --the number of satellites on-orbit for which the probability is being updated.
5. $T2$ --a temporary variable for computing current time of the simulation.

For example, members of $\{\lambda_n(t)\}$ or $\{\mu_n(t)\}$ which are dependent upon a specific n must be related through software logic to the program variable I . If a member of the above parameter set is time dependent, then the software logic must compute values of L or MUT dependent upon the value of $T2$ computed by

$$(2.31) \quad X = J$$

$$(2.32) \quad T2 = T1 + (X - 1.0) \cdot D$$

Examples of the uses of these values in computing members of the parameter sets for modeling purposes appear in Section 4. The typical modeler analyzing satellite systems will *only* have to modify the coefficient set block to measure the impact of various launch or orbital activities on satellite effectiveness.

The rate of change block updates the values of $dP(n)_t$ for a given value of n (software variable I) and t (software variable $T1$). These equations are simply the implementation of (1.1) to (1.4) using the current value of L and MUT as the appropriate value of $\lambda_n(t)$ and $\mu_n(t)$. The rate of change is stored as program value F and is inserted into (2.23) which is implemented in subroutine INTEG. A

tail block is necessary since this is a FORTRAN subroutine and consists of separate RETURN and END statements. For an example of subroutine FNS with the blocks delineated, see Fig. 1.

Modelling the effects of transient phenomena of launch and satellite operations is thus reduced to simple logic changes within the coefficient set block of subroutine FNS. The next few sections demonstrate how various phenomena can be modelled through manipulation of these coefficients.

3. THE INFINITE INVENTORY CASE

Section 2.0 presented a variety of methods for solving equations (1.1) to (1.4). The result was to adopt the method of continuous simulation due to its flexibility in modelling a range of transient phenomena. This section attempts to validate the continuous simulation approach. A simple situation is formulated for which a closed form solution exists. The simulation and closed form solutions are then compared. Finally, the limitations of the simple situation are explicated.

Consider a satellite system which maintains at most M satellites on-orbit. Satellites have a lifetime distribution with an average lifetime summarized by a number called Mean Mission Duration (MMD). Replenishment occurs on a probabilistic basis with an average time between launches characterized by a single number described as Mean Time Between Launches (MTBL). Note that MTBL is a summary measure of the performance of the launch system.

In fact, the satellite/launch system described above is the general form of the system described in Section 1. The repetition of satellite failures and replenishment launches with no consideration for number of satellites or launch vehicles procured implies an infinite number of each being procured. Thus, this case is referred to as the infinite inventory case. The surrealistic nature of this case is described in Section 3.3.

ACTUAL CODE SEGMENT

BLOCK NAME

SUBROUTINE FNS		HEADING
INTEGER M,NS REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX REAL P(75),E(500),P1(75),T(500),T2,F,MUT INTEGER I,J,K		TYPE DECLARATION
COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I		COMMON
C	MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500	COEFFICIENT SET
C		
C		
C		
C	COMPUTE TIME DEPENDENT PARAMETERS	
C		
	X=J	
	T2=T1+(X-1.0)*D	
	IF(T2.GT((M-C)/(C*MU))	
C		
C	STILL IN LAUNCH REPLENISHMENT PHASE	RATE OF CHANGE
C		
	L= (1.0/MMD)	
	MUT=C* MU	
	GO TO 5	
C		
C	ALL REPLENISHMENTS DEPLOYED	
C		
4	L=1.0/MMD	
	MUT = 0	
C	CHECK TO SEE NO. ON ORBIT FOR WHICH PROB. IS	RATE OF CHANGE
C	BEING UPDATED. THIS DETERMINES WHICH EQUATION IS	
C	BEING COMPUTED.	
C		
5	IF (I.GT.1) GO TO 1	
	F=-MUT*P1(I)+L*P1(I+1)	
	GO TO 3	
1	IF (I.GE.C+1) GO TO 2	
	X=I	
	F=-((X-1.0)*L+MUT)*P1(I)+X*L*P1(I+1)	
	GO TO 3	
2	IF (I.EQ.M+1) GO TO 6	RATE OF CHANGE
	I1=C	
	X=I	
	F=-((X-1.0)*L+MUT)*P1(I)+MUT*P1(I-I1)+X*L*P1(I+1)	
	GO TO 3	
6	I1=C	
	X=I-1	
	F=-X*L*P1(I)+MUT*P1(I-I1)	
3	CONTINUE	
	RETURN	
	END	TAIL

Fig. 1--Basic blocks within subroutine FNS

3.1 Closed Form Solution

For the limited case described above, a convenient closed form solution is available. In order to use the solution, the parameters of the solution must be described in terms of the performance measures MMD and MTBL.

Let the members of the parameter sets $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ be described by the relations:

$$(3.1) \quad \lambda_n(t) = \lambda = 1/\text{MMD} \quad \text{for all } n, t$$

$$(3.2) \quad \mu_n(t) = \mu = C/\text{MTBL} \quad \text{for all } n, t$$

$$(3.3) \quad a = \text{number on-orbit at time } t = 0$$

In this case, conditions (1.1) to (1.4) reduce to the following equations:

$$(3.4) \quad \frac{dP_0(t)}{dt} = -\mu P_0(t) + \lambda P_1(t)$$

$$(3.5) \quad \frac{dP_n(t)}{dt} = -(n\lambda + \mu)P_n(t) + \mu P_{n-1}(t) + (n+1)\lambda P_{n+1}(t) \quad 0 < n < M$$

$$(3.6) \quad \frac{dP_M(t)}{dt} = -M\lambda P_M(t) + \mu P_{M-1}(t)$$

$$(3.7) \quad P_n(0) = \begin{cases} 1 & \text{if } n = a \\ 0 & \text{if } n \neq a \end{cases}$$

This set of equations has a closed form solution described by Morse [3]. For $M > 0$, $a \geq 0$, the solution to (3.4) to (3.7) is described by

$$(3.8) \quad P_n^a(t) = P_n + \frac{2\rho^{1/2(n-a)}}{M+1} \sum_{s=1}^M \frac{\lambda}{\gamma_s} \left[\sin\left(\frac{s a \pi}{M+1}\right) - \sqrt{\rho} \sin\left(\frac{s(a+1)\pi}{M+1}\right) \right] \\ \cdot \left[\sin\left(\frac{s n \pi}{M+1}\right) - \sqrt{\rho} \sin\left(\frac{s(n+1)\pi}{M+1}\right) \right]$$

$$(3.9) \quad \gamma_s = \lambda + \mu - 2\sqrt{\lambda\mu} \cos\left(\frac{s\pi}{M+1}\right)$$

$$(3.10) \quad P_n = \left[\frac{1 - \rho}{1 - \rho^{M+1}} \right] \rho^n$$

$$(3.11) \quad \rho = \frac{\mu}{\lambda} \neq 1$$

$$(3.12) \quad \text{for } n = 0, \dots, M$$

Using (3.8) to (3.12), one can compute the required performance measures as follows:

$$(3.13) \quad A(t|a) = \sum_{n=N}^M P_n^a(t)$$

$$(3.14) \quad E[n|t, a] = \sum_{n=0}^M n P_n^a(t)$$

In (3.13), N equals the minimum number of active satellites required on-orbit to establish the availability of the satellite system. Additionally, Gross and Harris [2] give a solution to (3.8) to (3.12) for the case in which $M \rightarrow \infty$. The latter is of no interest for satellite systems, since this would allow an infinite number of satellites on-orbit.

3.2 Comparison With SPACE Model

A typical satellite system requires relatively few satellites on-orbit. Suppose a satellite system requires at most $M = 3$ satellites on-orbit. Generally, satellites have reasonably long lifetimes. This prototype is assumed to have an MMD = seven years = 84 months. A desired goal is to achieve orbital configurations relatively quickly. Therefore, an MTBL of one month is assumed (recall the assumption that this pattern repeats indefinitely). Since the current launch mechanism is by expendable vehicle (Titan, Delta, etc.), only one satellite is to be launched per launch vehicle. Finally, the counting process ($t = 0$) begins with no satellites on-orbit.

In terms of the parameters required for both the simulation and closed form solutions, the preceding description may be summarized as follows:

$$(3.15) \quad M = 3$$

$$(3.16) \quad \mu = C/MTBL = 1/1 \text{ month} = 1/\text{month}$$

$$(3.17) \quad \lambda = 1/MMD = 1/84 \text{ months} = .012/\text{month}$$

$$(3.18) \quad a = 0$$

$$(3.19) \quad c = 1 \text{ satellite/launch}$$

All of these parameters are used in both the closed form solution (3. 8) to (3.12) and in the continuous simulation model SPACE. The average number on-orbit over time is computed based upon the appropriate equations (for each approach) as described in earlier sections. The results are graphically compared in Fig. 2.

Several observations can be made regarding these results. The simulated and actual values are very close. In fact, the curves could be made to exactly overlap through appropriate choice of simulation parameters such as NS (the number of subintervals within a single time

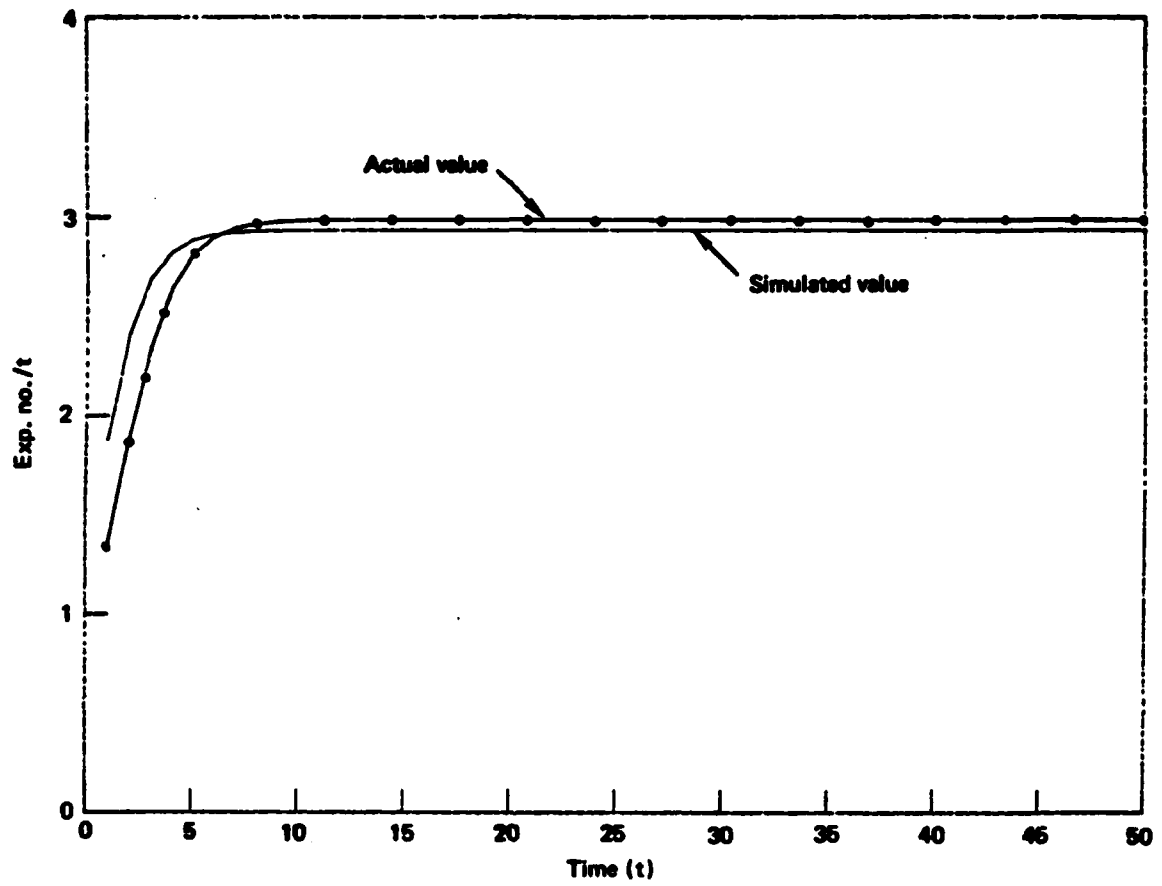


Fig. 2--Comparison of simulated and actual values

interval). The slight difference is maintained here simply to amplify the proximity between the simulated values and the actual values. From the perspective of the satellite system, both approaches demonstrate that the satellite system achieves its maximum quite rapidly (within six to eight months, depending upon the approach) and then settles down to a stable configuration. The time period prior to this stabilization is referred to as a transient state period. The remainder of the satellite system's operational cycle is quite stable and is characterized by the term steady state. Most satellite programs are greatly concerned about this transient phase, since this period determines the future success of the program.

The comparison is based upon a small number of satellites ($M = 3$) for reasons of mathematical tractability. In the typical application, the ratio of $\rho = \mu/\lambda$ is usually bounded by $0 \leq \rho \leq 1$, allowing M to range freely. The special characteristics of orbital/launch systems forces $\rho > 1$, requiring M to be bounded. The exact bounding value of M depends on the value of ρ and is not presented here.

3.3 Limitations of the Simple Example

Many of the reasons for the nonrealistic nature of this example have been discussed in Section 2.1. These are summarized and expanded somewhat in this section.

As demonstrated above, the transient phenomena are short-lived in this example. This is due to the assumption of the constant failure and replenishment rates in (3.16) and (3.17). Practical experience dictates that this may not be realistic, since the extent and range of transient phenomena are often more elaborate. The additional assumptions regarding the availability of an infinite number of both launch vehicles and spacecraft are totally inconsistent with traditionally small procurements of satellite programs. The sole purpose of using the closed form in this section is to allow a comparative evaluation of the continuous simulation implemented in model SPACE.

The next section of this Note uses the SPACE model to extend the analysis of satellite systems to include nonconstant failure and replenishment rates. Specific examples are presented demonstrating how the choice of $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ may reflect a variety of transient phenomena.

4. SPECIAL CASES

This section presents a series of examples which demonstrate the appropriate choice of members of $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ for several realistic operational philosophies. The first example focuses upon using a power cycling approach for operating a satellite in the context of a single lot buy. Another relevant example appears in the second section which investigates the impact of disruptions in launch activities. Two cases are presented--one in which no backup launch system is available and a second case in which a backup system with lesser capability is assumed to exist. Finally, a third section discusses the use of the results in life-cycle cost analysis using some of the outputs of the SPACE model. This last section also discusses (and presents contrived examples of) some criteria for using the SPACE model results in cost-effective policy decisions.

4.1 Satellite Power Cycling

In this section, two major aspects of satellite/launch operations are modelled. From the perspective of satellite orbital operations, the concept of power cycling is explored. Launch operations are modelled under the assumption that a single procurement of satellites is launched at a constant rate until all satellites are launched. After that point, no more launches occur dictating decay in the number of satellites on-orbit. The members of $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$ are fully described below and are all functions of parameters which are easily recognized from earlier sections.

Power cycling is defined as using the satellite at full-power for some fixed portion of time and then reducing power for another fixed time period. The cycling aspect occurs because the high-low pattern is repeated over the operational cycle of each individual satellite. During the periods of lower power level, the failure rate should be correspondingly lower. The failure rates (members of $\{\lambda_n(t)\}$) in the example for this section assume the following form:

$$(4.1) \quad \lambda_n(t) = \begin{cases} \lambda_1 = 1/MMD & T1 \leq t < T1 + (1/30)LT; \forall n \\ \lambda_2 = .15(1/MMD) & T1 + (1/30)LT \leq t \leq T1 + LT; \forall n \end{cases}$$

where MMD = Mean Mission Duration of a satellite,

T1 = time counter for tracking the last occurrence of a simulation output report,

LT = the length of a simulation interval.

Appearance of the constant .15 assumes that failures occur during low power at 15 percent of the active failure rate. Another assumption is that a satellite is at a high power level for the first 1/30th of a simulation interval and at low power for the remainder of the interval. Since LT is assumed to be one month in the example below, this assumption states that the satellite spends the first day of each month at high power and the remaining 29 days at low power.

Description of the replenishment or launch rates is dependent upon the belief that all satellites procured are launched, after which no launches occur. The corresponding members of $\{\mu_n(t)\}$ are described as follows:

$$(4.2) \quad \mu_n(t) = \begin{cases} \mu_1 = C/MTBL = C\mu & t \leq [(M-C)/C\mu]; \forall n \\ \mu_2 = 0 & [(M-C)/C\mu] < t; \forall n \end{cases}$$

where C = number of satellites/launch vehicle,

MTBL = mean time between launches,

M = number of satellites procured,

μ = average number satellites launched per time unit.

The time required to launch M satellites at a rate of $C\mu$, conditioned upon the 1st launch occurring at time 0 is simply $[(M-C)/C\mu]$. Thus, the launch rate for this time period is simply $C\mu$ (which is the number of satellites per launch divided by the mean time between launches). After $[(M-C)/C\mu]$ time has passed, no more launches occur; therefore, the replenishment rate is reduced to zero.

Constructing the parameter set block of subroutine FNS is a simple task. Once this routine has been appropriately modified, the SPACE model is executed with the following inputs:

$$(4.3) \quad \text{MMD} = 120 \text{ months}$$

$$(4.4) \quad C = 1/\text{launch}$$

$$(4.5) \quad M = 10$$

$$(4.6) \quad \text{MTBL} = 6 \text{ months}$$

The same input data was used in a version of subroutine FNS in which no cycling was allowed. In other words, the set $\{\lambda_n(t)\}$ was represented as

$$(4.7) \quad \lambda_n(t) = \lambda = \frac{1}{\text{MMD}} \text{ for all } n, t$$

The results of using both (4.1) and (4.7) appear in Fig. 3 and are discussed in the next paragraph. Both models assume that one percent of all launches fail to achieve orbit. Both systems obtain the peak number on-orbit at approximately the same time. Some careful thought reveals that these peaks appear when replenishments cease to occur-- at time $[(M-C)/C\mu]$. Using the data from (4.3) to (4.6), $\mu = C/\text{MTBL} = 1/6$ per month so that the peaks occur at roughly time $[(M-C)/C\mu] = 54$ months. Figure 3 does indeed verify this number.

A comparison of the impacts of power cycling can now be made. Cycling achieves a higher peak level of satellites on-orbit because satellites are not failing as often. Moreover, satellites operated on a power-cycling basis obtain more satellites on-orbit earlier than full-power satellites and maintain more satellites on-orbit for a significantly longer time period. Of course, the actual construction of these curves is dependent upon the inputs in (4.3) to (4.6). However,

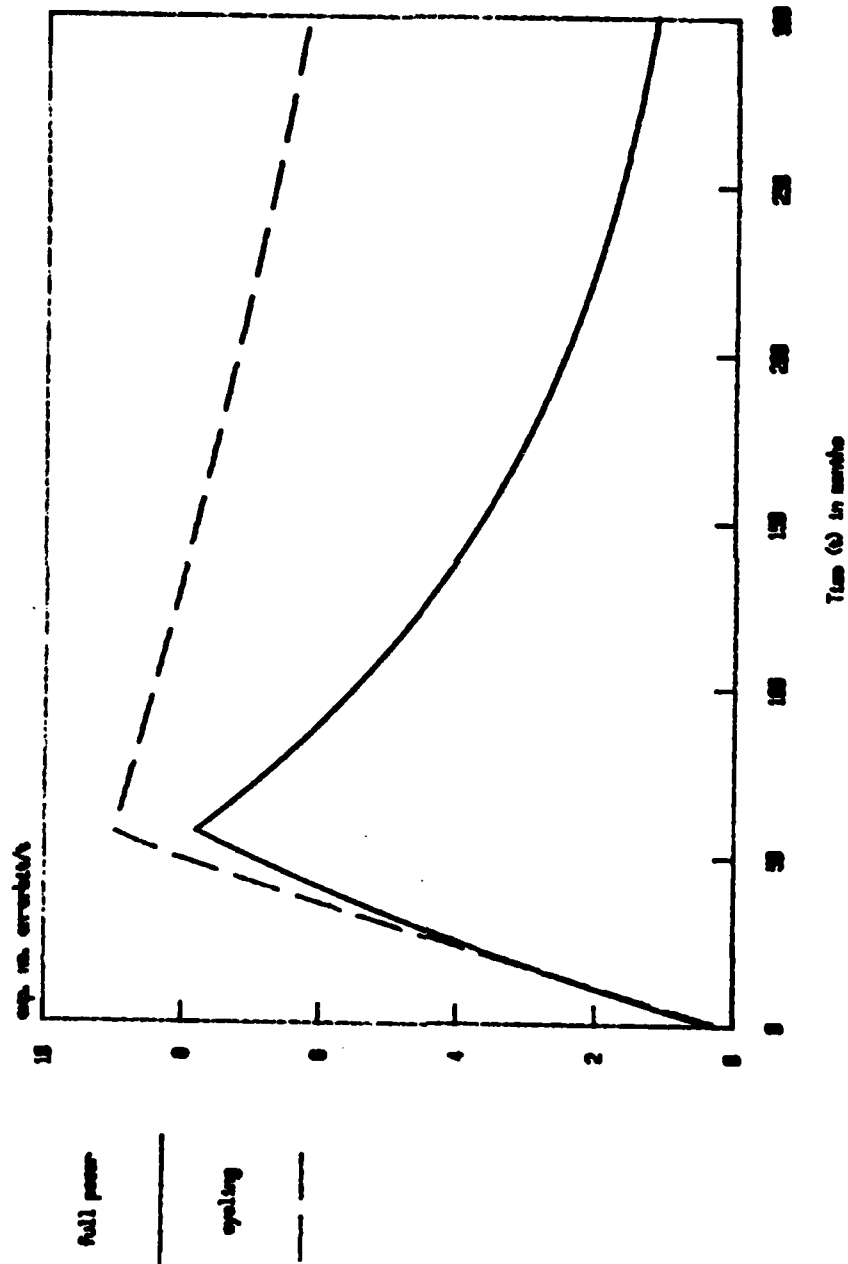


Fig. 3--Power cycling vs fully powered

intuition indicates that the relationships and conclusions will still be the same regardless of the actual inputs; the magnitudes of the differences between the curves in Fig. 2 would vary for different inputs.

More importantly, Fig. 2 demonstrates that the modelling approach employed in (1.1) to (1.4) and implemented through continuous simulation is quite capable of representing the residual lifetimes of the satellites on-orbit. In fact, the downward sloping portions of the curves in Fig. 2 are simply the decaying lifetimes of the satellites currently on-orbit, since no launches or replenishments occur. Having effectively analyzed satellite operations, the next section approaches the process of detailed analysis of launch operations.

4.2 Disruptions in Launch Operations

Since the focus of this paper is specifically to relate launch operations to satellite performance, this example contrasts two situations which reflect varying launch conditions. At first, a 36-month launch disruption is assumed. After the 36 months have passed, a new launch system becomes operational with significantly degraded capability. The analogue is to consider a complete failure of the shuttle system with *no* operational backup system of expendables. In contrast, the same analysis is performed under the assumption that a backup system of expendables (with degraded capability) is immediately available. Failure modes are of no particular interest in these two analyses. Therefore, the following failure parameters are assumed throughout this section:

$$(4.8) \quad \lambda_n(t) = \lambda = 1/MMD \quad \forall n, t$$

In order to allow comparative statements regarding the impact of differing launch scenarios, all other periods of operation are assumed to be steady state. This translates to allowing an infinite number of launches and to accepting an infinite inventory of satellites. Thus, *M* is now interpreted as the maximum number of satellites allowed

on-orbit. The construction of equations (1.1) to (1.4) is such that a launch is conditioned upon having less than M satellites on-orbit, and no launches are allowed if 10 are on-orbit.

For the no-backup case, consider that the space shuttle is the primary launch mechanism. At time $t = 25$, the shuttle becomes disabled. If immediate actions are taken, the Titan system becomes operational 36 months later. However, in general, the Titan carries a lesser payload. The replenishment rates [members of $\{\mu_n(t)\}$] describing this no-backup case may be described by

$$(4.9) \quad \mu_n(t) = \begin{cases} \mu_1 = C(1/MTBL) = C\mu & t \leq 25; \forall n \\ \mu_2 = 0 & 25 < t < 61; \forall n \\ \mu_3 = .5C\mu & 61 \leq t \end{cases}$$

where C = number satellites/launch,

MTBL = mean time between launches,

μ = average number of launches per unit time.

Recall that the new launch system must have a degraded capability.

For a 32,000 pound payload, the shuttle can carry two satellites per launch while a Titan can launch only one satellite per vehicle. Thus, the degraded capability in this example is .5 the shuttle capability (hence, the appearance of the parameter .5 in μ_3).

As an alternative, one could maintain a backup system of expendable vehicles (at a cost, naturally) so that the loss of the shuttle system at time $t = 25$ results in an immediately degraded launch capability with no complete disruption in launches. For the backup case, the appropriate coefficients are described as:

$$(4.10) \quad \mu_n(t) = \begin{cases} \mu_1 = C(1/MTBL) = C\mu & t \leq 25; \forall n \\ \mu_2 = .5C\mu & 25 < t; \forall n \end{cases}$$

where C, MTBL, and μ are described above. Justification for use of the parameter .5 is the same as explained earlier.

After subroutine FNS was modified to reflect (4.9), the SPACE model was executed using the following input set:

(4.11) MMD = 120 months

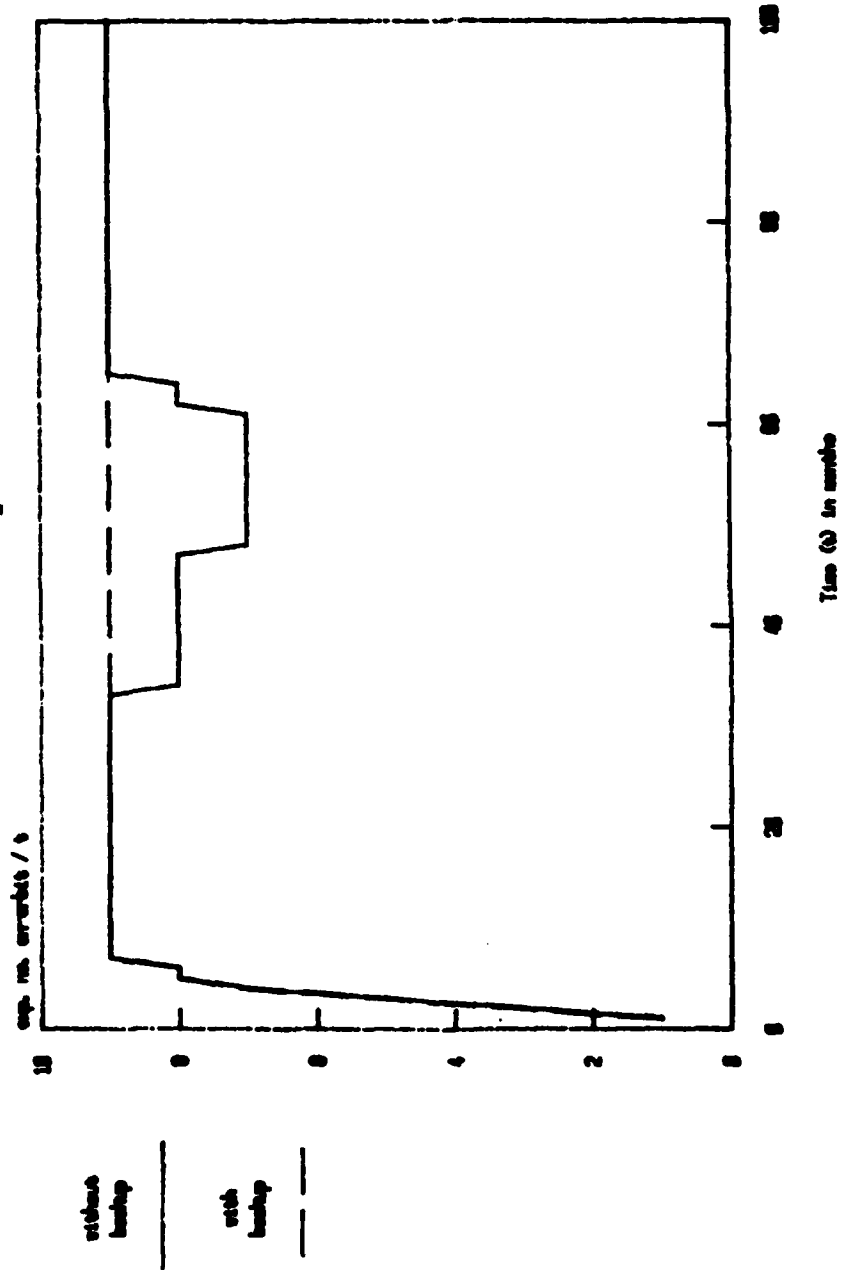
(4.12) C = two satellites/launch

(4.13) MTBL = one month between launches

The results appear in Fig. 4 labeled by the title "Without Backup." Similar steps were taken for (4.10) and SPACE was executed again using (4.11) to (4.13). These results also appear in Fig. 4 labeled "With Backup."

The curves in Fig. 4 lead to several important conclusions. For purposes of this discussion, assume that the operational goal is to maintain nine active satellites on-orbit. In the no-backup case, the disruption at $t = 25$ does not have an immediate impact due to the residual lifetimes of the satellites. Orbital decay begins to degrade the satellite configuration at $t = 34$, so that, on the average, this system can sustain operations for about nine months without replenishment. Moreover, while the new launch system is introduced at $t = 61$, the orbital configuration is not fully recovered until $t = 68$ (seven months later). Thus, the satellite system is "unavailable" or "down" (assuming nine satellites are absolutely required) for $68 - 34 = 34$ months. The down time is less than 36 months due to the slight compensating effect of the residual lifetimes of the satellites.

Now, consider the case where the backup system was available. Fig. 4 reveals that although the backup system introduced at $t = 25$ had only half the capability of the shuttle system, the immediate employment of the backup in combination with the satellite residual lifetimes resulted in *no loss* of active satellites on-orbit. Note also that prior to and subsequent to the launch activity variations, both curves are identical. Moreover, the effective slope of the initial



Note: Granularity of the plotter package was maintained at the integer level for purposes of exposition.

Fig. 4--Effects of launch disruption

deployment activities ($t = 0$ to $t = 11$) is roughly twice the effective slope in the no-backup case when the new launch system is introduced ($t = 61$ to $t = 68$). This reflects the .5 lesser capability of the backup system.

As a result, one can say that if the backup system is procured and available for immediate employment then the mission can be performed with little or no time delay (remember, these curves represent averages, so some probability exists that a delay does occur). The final subsection deals with relating this information to cost-effective policy decisions.

4.3 The SPACE Model and Cost-Effectiveness

The information throughout this Note can now be used to make cost-effective policy decisions. In the last section, some conclusions were made regarding a variety of launch scenarios. Measures of both cost and effectiveness can be employed to evaluate the implications of these conclusions. This section suggests an appropriate cost measure, several possible effectiveness measures, and gives an example of a cost-effectiveness comparison.

An extension of the life-cycle cost formula in [1] can now be stated. This extension has the added attraction that discounting can now be specifically included since the cost measure is now a time profile (due to its dependence on $P_n(t)$). Life cycle costs for a satellite system can now be approximated by

$$(4.14) \quad LCC(t) = RDTE + C_p \sum_{n=0}^M P_n(t) \mu_n(t) e^{-rt} + C_L \sum_{n=0}^M P_n(t) \mu_n(t) e^{-rt} \\ + C_I \sum_{n=0}^M P_n(t) [\mu_n(t) - \lambda_n(t)] e^{-rt} + C_B e^{-rt}$$

with total expected, discounted life cycle costs easily computed as

$$(4.15) \quad E(LCC) = \sum_{t=0}^{LC} LCC(t)$$

where M = the maximum number of satellites,
 $\{\lambda_n(t)\}$ = the set of failure rates,
 $\{\mu_n(t)\}$ = the set of replenishment rates,
 e^{-rt} = the infinitesimal discounting factor
 r = the discount rate

RDTE = the research, development, testing, and evaluation costs,

C_p = production cost/satellite,

C_L = launch cost/satellite,

C_I = inventory carrying cost/satellite/time unit,

C_B = backup launch system costs/time unit,

LC = the life cycle of the satellite system.

Each term is a separate cost incurred for the various recognized phases of the life cycle. Discounting may now be employed since costs are a function of time through the factors $P_n(t)$. The factor $\mu_n(t) - \lambda_n(t)$ represents the rate at which inventory accumulates at time t . An additional module can easily be added to the SPACE model to compute and output the value for $LCC(t)$ in (4.14) when $P_n(t)$ is outputted. The coding logic of this module must determine when and how to add the various costs. For instance, if the backup system is not operational, $C_B > 0$. However, should the backup system become operational, $C_B = 0$ and C_L should be modified to reflect the launch costs of the new system.

Effectiveness is easily measured through the satellite performance measures--availability $[A(t)]$ or average number on-orbit $(E[n|t])$. Using either of these, the operational goal is to maintain a minimum number on-orbit over a time period T or to maintain a minimum availability over time period T . (Recall, these two were shown to be roughly equivalent, in the sense that both summarize the same information.)

However, two other effectiveness criteria may prove to be interesting. Both of these criteria are based upon explicitly measuring the risk associated with a launch/orbital strategy. First, one may

desire to obtain a high $P_N(T)$ for some fixed number of satellites N and some fixed time T . Another measure of risk is the down time associated with a satellite system. Down time may be defined as the amount of time between the decay of an orbital configuration until its restoration (measured relative to a requirement to maintain N satellites on-orbit). For example, in the previous section, loss of the shuttle system for 36 months resulted in a down time of 34 months. Both of the latter measures are, in essence, inverse images of the first two measures. Maintaining a large average number of satellites on-orbit is equivalent to obtaining a significant $P_N(T)$ for N , T fixed.

Proper usage of results from the SPACE model dictates that one compare the impacts of variations in launch scenarios with the corresponding cost values. As an example, consider the chart in Table 1 below. This table represents the results of comparing the backup and no-backup cases for the shuttle launch system. The measure $P_N(T)$, for fixed N , T , is chosen as the appropriate effectiveness measure for two reasons. Risk is assumed to be of primary importance (depending upon the satellite mission). Moreover, a single value is useful for comparison against the single, average value for $E(LCC)$. Recall that the previous example required nine satellites to ensure mission performance. Risk is assessed at time $t = 60$ because this is during the potential launch disruption when backup launch facilities are most critical.

Table 1

COST-EFFECTIVENESS OF LAUNCH SYSTEM BACKUP

<u>Scenario</u>	<u>$P_9(60)$</u>	<u>$E(LCC)$</u>
Shuttle without backup	.6	\$400B
Shuttle with expendable backup	.9	\$1,000B

The costs in Table 1 are not accurate, but do reflect the higher costs associated with maintaining an expendable backup system. The probabilities were taken from actual computer runs. However, if the costs were to prove accurate, one could ascertain that a high risk (very low probability) is associated with having no backup launch system. Of course, the actual decision made depends upon such subjective factors as mission criticality, the actual cost magnitudes, and many individuals' attitudes regarding risk. If the space mission of the United States is highly critical for survival, then maintenance of the expendable backup system is absolutely necessary to achieve the lower level of risk (reflected by higher $P_0(60)$ in Table 1).

As explained, these figures are estimates only. The costs have not been computed. In addition, more background is required concerning the nature of the appropriate backup system alternatives--their performances relative to the shuttle and their respective yearly cost expenditures. Finally, an appropriate effectiveness measure must be chosen.

5. CONCLUSIONS

The conclusions section of reference [1] suggested several extensions of that work. In this Note, the extensions have been accomplished with the addition of the explicit focus upon analysis of space shuttle activities. A useful tool has been provided for analyzing transient phenomena in the operation of space systems--both from the orbital and launch perspectives. More importantly, measures have been derived which allow the comparison of a variety of launch scenarios.

The continuous simulation tools described in this Note and its Appendixes are simple to implement. They may be programmed on a hand-held calculator and provide significant information based upon a minimal amount of data. System state changes are easily modelled by a system of differential equations. The most difficult task in applying the tools to satellite systems is in the appropriate choice of failure and replenishment rates--the members of $\{\lambda_n(t)\}$ and $\{\mu_n(t)\}$, respectively.

More importantly, this Note has demonstrated that an expendable backup system for the space shuttle can have significant positive effects upon the U.S. space posture. Reasonably exact measures of risk and cost can be assessed with minimal effort, limited only by the accuracy of the tools used. The utility of this kind of information in making long-term policy decisions is high.

Appendix A

USER INSTRUCTIONS

1. Decide the forms required for $\{\mu_n(t)\}$, $\{\lambda_n(t)\}$.
2. Under TSO, use the editor to modify the coefficient set block of subroutine FNS. The whole SPACE model is stored in the partitioned data set member named CSIMLIB.SPACE.FORT (BUNCH).
3. Compile CSIMLIB.SPACE.FORT(BUNCH) into CSIMLIB.SPACE.OBJ(BUNCH).
4. Create two output files:
 - a. CSIMLIB.SPACE.OUT(SUMRY): lists the expected number on-orbit over time.
 - b. CSIMLIB.SPACE.OUT(DETAIL): lists the $P_n(t)$ and cumulatives over time.
5. To execute, type the following:
ALLOC F(FT09F001) DA(CSIMLIB.SPACE.OUT(SUMRY))
ALLOC F(FT10F001) DA(CSIMLIB.SPACE.OUT(DETAIL))
LOADGO CSIMLIB.SPACE.OBJ(BUNCH)
6. The computer will prompt you for the inputs, as follows:
INPUT M, P0, MMD, C, MU
? 10, .01, 120, 1, 1 USER INPUTS VALUES
INPUT NO. SUBINT AND LENGTH OF TIME INTERVAL
? 100, 1 USER INPUTS VALUES
INPUT T1, PMIN, TMAX
? 0, .9, 300 USER INPUTS VALUES
(Some time passes during execution; time intervals printed on CRT.)
INPUT 1 FOR TITLES AND DETAILED INFO
? 1 USER TYPES 1 TO ECHO INPUT VALUES
7. Outputs are now stored in dataset members described in 4) above. Send copy to line printer or use plotter to prepare graphs.

Appendix B

MODEL DOCUMENTATION

A. Subroutine descriptions

1. READ--reads the input data.
2. INTEG--performs the numerical integration process.
3. WRITE--writes the outputs $[E[n|t], P_n(t)]$ to the appropriate disk files.
4. AVG--computes the average number at time t .
5. FNS--contains the implementation of equations (1.1) to (1.4) which form the rates of change of the probabilities.

B. Input variables

1. P_0 --probability that a booster fails.
2. MMD--mean mission duration = average life of a satellite
(= $1/\text{average number failures per unit time} = 1/\text{failure rate}$).
3. C--number satellites/launch vehicle.
4. MU--average number launches/unit time = $(1/\text{Mean Time Between Launches}) : (1/\text{MTBL}) = \text{launch rate}$.
5. M--maximum number of satellites on-orbit (i.e., number satellites procured).
6. LT--length of the simulation time interval (in time units consistent with $1/\text{MU}$ and MMD above).
7. NS--number of subintervals within time interval LT.
8. T_0 --the time for initialization of the simulation. If $T_0 = 0$, a special integration step computes $P_n(T_0)$.
9. PMIN--a stopping criterion which halts the simulation process if $P_0(t)$ is smaller.
10. TMAX--another stopping criterion which is interpreted as the maximum number of increments of size LT for simulating.

C. Output variables

1. D = length of a subinterval = LT/NS
2. $T1$ = time counter for endpoints of simulation intervals (time at which simulation measures are output).
3. L = failure rate (computed in FNS as a function of MMD , n , t).
4. MUT = replenishment rate (computed in FNS as a function of $MTBL$, n , t).
5. J = program subinterval counter within an integration interval.
6. F = program counter for number on-orbit ($I = 1$ is equivalent to $n = 0$).
7. $T2$ = temporary variable for storage of actual time associated with subinterval J .
8. F = value for $dP_n(t)/dt$, evaluated within subroutine FNS.
9. $P(I)$ = probability that $(I-1)$ satellites are on-orbit at a given time $T1$.
10. $CP(I)$ = cumulative probability that less than or equal to $(I-1)$ satellites are on-orbit at a given time $T1$.
11. $Pl(I)$ = updated probability that $(I-1)$ satellites are on-orbit at a given time $T2$.
12. $E(K)$ = expected number on-orbit at K th time point.
13. $T(K)$ = the time at K th time point.

Appendix C
"SPACE" MODEL SOFTWARE

INTEGER M,NS

REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX

REAL P(75),E(500),P1(75),T(500),T2,F,MUT

INTEGER I,J,K

COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX

COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I

C

C

MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500

C

CALL READ

C

CALL INTEG

C

CALL WRITE

C

STOP

END

SUBROUTINE AVG

INTEGER M,NS

REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX

REAL P(75),E(500),P1(75),T(500),T2,F,MUT

INTEGER I,J,K

COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX

COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I

C

C

MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500

C

C

C

COMPUTE THE AVERAGE NO. ON ORBIT AT TIME T1

C

C

C

COMPUTE THE UPPER LIMIT OF THE DO-LOOP

C

JJ=M+1

C

C

COMPUTE THE AVERAGE

C

DO 1 I1=1,JJ

1

E(K)=E(K)+(I1-1)*P(I1)

C

RETURN

END

SUBROUTINE READ

INTEGER M,NS

REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX

REAL P(75),E(500),P1(75),T(500),T2,F,MUT

INTEGER I,J,K

REAL LT1,NS1

COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX

COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I

C

C MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500

C

WRITE(6,1)

1 FORMAT (1X,'INPUT M, P0, MMD, C, MU ')

READ(5,*) M,P0,MMD,C,MU

WRITE(6,2)

2 FORMAT(1X,'INPUT NO. SUBINT AND LENGTH OF TIME INTERVAL')

READ(5,*)NS,LT

WRITE(6,7)

7 FORMAT(1X,'INPUT T1 , PMIN , TMAX ')

READ(5,*) T1,PMIN,TMAX

C

C CHECK TO SEE IF NO. SUBINTERVALS >= 100

C

IF (NS.GE.100) GO TO 5

WRITE(6,6)

6 FORMAT(1X,'NS < 100 . TRY AGAIN')

STOP

C

C CHECK TO MAKE SURE $C*MU < NS/LT$ FOR INTEGRATION TO HOLD

C

5 IF ($C*MU.LT.NS/LT$) GO TO 3

WRITE(6,4)

4 FORMAT(1X,'C*MU NOT < NS/LT AS REQUIRED FOR INTEGRATION'/)

STOP

C

C COMPUTE NO. OF SUBINTERVALS PER TIME INTERVAL

C

3 $LT1=LT$

$NS1=NS$

$D=LT1/NS1$

C

C

RETURN

END

SUBROUTINE WRITE

INTEGER IE(500)

INTEGER M,NS

REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX

REAL P(75),E(500),P1(75),T(500),T2,F,MUT

INTEGER I,J,K

COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX

COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I

C

C MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500

C

C

WRITE(6,15)

15 FORMAT(/1X,'INPUT 1 FOR TITLES AND DETAILED INFO'/)

READ(5,*)IS

IF(IS.NE.1) GO TO 16

WRITE(9,13)M

13 FORMAT(/1X,'MAX. NO. SATELLITES = ',I5)

WRITE(9,9)P0

9 FORMAT(/1X,'PROB. OF BOOSTER FAILURE = ',F6.3)

WRITE(9,10)MMD

10 FORMAT(/1X,'MEAN MISSION DURATION = ',F7.2)


```
WRITE(9,11)C
11  FORMAT(/1X,'NO. SATELLITES PER LAUNCH = ',F6.0)
WRITE(9,12)MU
12  FORMAT(/1X,'NO. LAUNCHES PER UNIT TIME = ',F6.2//)
C
C
WRITE(9,1)
1   FORMAT (1X,'EXPECTED NO. ON ORBIT AS A FUNCTION OF TIME')
WRITE(9,2)
2   FORMAT(/1X,'TIME',5X,'EXPECTED NO.'//)
16  CONTINUE
DO 3 I=1,K
    IE(I)=E(I)
3   WRITE(9,4) T(I),IE(I)
4   FORMAT(1X,F6.2,6X,I8)
    IF(IS.NE.1) GO TO 17
WRITE(9,5) T(K)
5   FORMAT(/1X,'PROB. DIST. OF NO. ON ORBIT AT TIME ',F10.2)
WRITE(9,8)
8   FORMAT(/1X,'NUMBER',5X,'PROBABILITY',/)
    JJ=M+1
DO 6 I=1,JJ
```

JJ1=I-1

6 WRITE(9,7) JJ1,P(I)
7 FORMAT(2X,I4,9X,F6.4)
17 CONTINUE

 RETURN

 END

SUBROUTINE INTEG

INTEGER M,NS

REAL CP(75)

REAL PO,MMD,C,MU,LT,D,L,T1,PMIN,TMAX

REAL P(75),E(500),P1(75),T(500),T2,F,MUT

INTEGER I,J,K

REAL NS1,CO,MUO,MO

COMMON /INPUTS/M,PO,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX

COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I

C

C

MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500

C

C

C

REDUCE MAGNITUDE FOR EASE OF COMPUTATION

C

FR=-1.0

IF (C.EQ.1.0.AND.MU.EQ.1.0) GO TO 14

C

C

STORE C,MU,M INTO TEMPORARY STORAGE LOCATIONS

C

CO=C

MUO=MU

M0=M

C=1.0

MU=1.0

R6=M/(C0*MU0)

M=R6

FR=R6-M

C

C CHECK FOR SPARSENESS OF DATA

C

14 IF (M.LE.75) GO TO 15

WRITE(6,16)

16 FORMAT(1X,'TOO LITTLE DATA FOR SUCH LARGE NUMBERS')

STOP

C

C INITIALIZE UPPER LIMIT ON DO LOOPS

C

15 JJ=M+1

C

C INITIALIZE THE PROBABILITIES

C

C THE INITIALIZATION SCHEME BELOW ASSUMES THAT TIME

C ZERO IS EQUIVALENT TO THE TIME OF THE FIRST LAUNCH

```
C      IN THE SYSTEM.
C
      DO 2 I=1,JJ
      IF (I.EQ.C+1) P(I)=1-P0
      IF (I.NE.C+1) P(I)=0.0
2      CONTINUE
C
C      INITIALIZE COUNTER FOR NO. OF TIME INCREMENTS EVALUATED
C
      K=0
C
C      BEGIN THE TIME ITERATION PROCESS
C
C      IF T1 GT 0 THEN TREAT 0 TO T1 AS FIRST TIME INTERVAL
C
      J1=1
      IF (T1.EQ.0) GO TO 1
      J1=0
C
C      RECORD TIME SUBINTERVAL D INTO TEMPORARY STORAGE D1
C
      D1 = D
```

D2=NS

C

C COMPUTE SPECIAL D FOR INTERVAL FROM 0 TO T1 > 0

C

NS1=5000.0

NS=NS1

D=T1 / NS1

GO TO 7

C

1 T1=T1 + LT

7 K=K+1

C

C SINCE TIME INCREMENTS MAY NOT BE INTEGRAL, CREATE A VECTOR

C TO STORE TIME VALUES. INITIALIZE EXPECTED NO. ON ORBIT

C FOR A GIVEN TIME LOCATION.

C

T(K)=T1

E(K)=0.0

C

C BEGIN THE INTEGRATION BY STORING PROB. AT TIME T1 INTO

C A TEMPORARY STORAGE VECTOR.

C

```
DO 3 I=1,JJ
3  P1(I)=P(I)
C
C  UPDATE THE TEMPORARY PROBABILITIES BY NUMERICALLY INTEGRATING
C  THROUGH A SINGLE TIME INCREMENT.  SIMPLE EULER METHOD OF
C  NUMERICAL INTEGRATION IS USED.
C
DO 4 J=1,NS
DO 4 I=1,JJ
CALL FNS
4  P1(I)=P1(I)+D*F
C
C  RESTORE INTEGRATED RESULTS FROM TEMPORARY PROB. VECTOR P1
C  TO PERMANENT PROB. VECTOR P TO CONINUE THE INTEGRATION
C  OVER THE NEXT TIME INTERVAL.
C
DO 5 I=1,JJ
5  P(I)=P1(I)
C
C  THE PROBABILITIES AS A FUNCTION OF TIME ARE NOT PERMANENTLY
C  STORED AFTER USE IN CALCULATING THE MEAN BELOW.  THUS, IF
C  THE ACTUAL PROBABILITIES ARE DESIRED, THIS IS THE PLACE
```

```
C      TO STORE OR OUTPUT THE PROBABILITIES.
C
      CALL AVG
C
C      IF THIS WAS SPECIAL CASE WHEN T1 > 0 RESTORE SUBINTERVAL
C      LENGTH FROM TEMPORARY STORAGE D1 TO ACTUAL VARIABLE D
C
      IF (J1.GT.0) GO TO 8
      J1=1
      D=D1
      NS=D2
8      CONTINUE
C
C      COMPUTE CUMULATIVES FOR OUTPUT
C
      DO 20 I=1,JJ
20      CP(I)=0.0
      CP(1)=P(1)
      DO 21 I=2,JJ
21      CP(I)=CP(I-1)+P(I)
      GO TO 100
C
```



```
C      OUTPUT  PROBABILITIES
C
      T2=T1-LT
      WRITE(10,9) T2
9      FORMAT(/1X,'PROB. VECTOR AT  TIME ',F7.2//)
      WRITE(10,*) (P(I),I=1,JJ)
      WRITE(10,*) (CP(I),I=1,JJ)
100     CONTINUE
C
C
C      OUTPUT MESSAGE TO TERMINAL FOR STATUS REPORT TO PROGRAMMER
C
      WRITE(6,12)T1
12      FORMAT(1X,'TIME = ',F7.2)
C
C
C
C      CHECK TO SEE IF MAX NO. OF TIME INTERVALS HAVE BEEN EVALUATED
C
      IF (K.GT.TMAX) GO TO 6
C
C      CONTINUE INTEGRATION FOR NEXT TIME INCREMENT
```

C CHECK TO SEE IF EXPECTED NO. ON ORBIT HAS REACHED ZERO

C

IF (P(1).GT.PMIN.OR.E(K).LT.1.0) GO TO 6

GO TO 1

6 CONTINUE

C

C RESTORE SCALE OF MAGNITUDE, IF NECESSARY

C

IF(FR.LT.0.0) GO TO 18

C

C=C0

MU=MU0

M=M0

C

C RESTORE SCALE TO EXPECTED VALUES

C

DO 17 I=1,K

17 E(I)=C*MU*(E(I)+FR)

18 CONTINUE

RETURN

END

SUBROUTINE FNS

INTEGER M,NS

REAL P0,MMD,C,MU,LT,D,L,T1,PMIN,TMAX

REAL P(75),E(500),P1(75),T(500),T2,F,MUT

INTEGER I,J,K

COMMON /INPUTS/M,P0,MMD,C,MU,NS,LT,D,T1,L,PMIN,TMAX

COMMON /OUTPUT/P,E,P1,T2,F,T,K,MUT,J,I

C

C MAX NO. SAT = 75 ; MAX NO. TIME PERIODS = 500

C

C

C

C COMPUTE TIME DEPENDENT PARAMETERS

C

 X=J

 T2=T1+(X-1.0)*D

 IF(T2.GT((M-C)/C*MU))) GO TO 4

C

C STILL IN LAUNCH REPLENISHMENT PHASE

C

 L= (1.0/MMD)

 MUT=C* MU

```
GO TO 5

C
C ALL REPLENISHMENTS DEPLOYED
C
4 L=1.0/MPD
  MUT = 0

C
C CHECK TO SEE NO. ON ORBIT FOR WHICH PROB. IS
C BEING UPDATED. THIS DETERMINES WHICH EQUATION IS
C BEING COMPUTED.
C
5 IF (I.GT.1) GO TO 1
  F=-MUT*P1(I)+L*P1(I+1)
  GO TO 3
1 IF (I.GE.C+1) GO TO 2
  X=I
  F=-((X-1.0)*L+MUT)*P1(I)+X*L*P1(I+1)
  GO TO 3
2 IF (I.EQ.M+1) GO TO 6
  I1=C
  X=I
  F=-((X-1.0)*L+MUT)*P1(I)+MUT*P1(I-I1)+X*L*P1(I+1)
```

GO TO 3

6 I1=C

X=I-1

F=-X*L*P1(I)+MUT*P1(I-I1)

3 CONTINUE

RETURN

END

REFERENCES

1. Krell, Bruce E., *Cost-Effectiveness Measures of Replenishment Strategies for Systems of Orbital Spacecraft*, The Rand Corporation, N-1295-AF, December 1979.
2. Gross, Donald, and C. M. Harris, *Fundamentals of Queueing Theory*, Wiley, 1974.
3. Morse, Philip M., *Queues, Inventories, and Maintenance*, Wiley, 1958.